```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBBBBBBBBBBB       AAA       AAA     SSSSSSSSS   RRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB       AAA       AAA     SSSSSSSSS   RRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB       AAA       AAA     SSSSSSSSS   RRRRRRRRRRR           TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR             TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR             TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR             TTT        LLL
BBB          BBB   AAA       AAA           SSS  RRR      RRR           TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR             TTT        LLL
BBB          BBB   AAA       AAA           SSS  RRR      RRR           TTT        LLL
BBB          BBB   AAA       AAA           SSS  RRR      RRR           TTT        LLL
BBB          BBB   AAA       AAA           SSS  RRR      RRR           TTT        LLL
BBBBBBBBBBBB       AAA       AAA   SSSSSSSSSSS   RRR         RRR       TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA       AAA   SSSSSSSSSSS   RRR         RRR       TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA       AAA   SSSSSSSSSSS   RRR         RRR       TTT        LLLLLLLLLLLLLLL
```

FILEID**BASPOWRR

```
BBBBBBBB      AAAAAA      SSSSSSSS  PPPPPPPP      000000    WW      WW  RRRRRRRR    RRRRRRRR
BBBBBBBB      AAAAAA      SSSSSSSS  PPPPPPPP      000000    WW      WW  RRRRRRRR    RRRRRRRR
BB      BB  AA      AA  SS        PP      PP  00      00  WW      WW  RR      RR  RR      RR
BB      BB  AA      AA  SS        PP      PP  00      00  WW      WW  RR      RR  RR      RR
BB      BB  AA      AA  SS        PP      PP  00      00  WW      WW  RR      RR  RR      RR
BB      BB  AA      AA  SS        PP      PP  00      00  WW      WW  RR      RR  RR      RR
BBBBBBBB    AA      AA    SSSSSS  PPPPPPPP    00      00  WW      WW  RRRRRRRR    RRRRRRRR
BBBBBBBB    AA      AA    SSSSSS  PPPPPPPP    00      00  WW      WW  RRRRRRRR    RRRRRRRR
BB      BB  AAAAAAAAAA          SS  PP        00      00  WW  WW  WW  RR  RR      RR  RR
BB      BB  AAAAAAAAAA          SS  PP        00      00  WW  WW  WW  RR  RR      RR  RR
BB      BB  AA      AA          SS  PP        00      00  WWWW  WWWW  RR    RR    RR    RR      ....
BB      BB  AA      AA          SS  PP        00      00  WWWW  WWWW  RR    RR    RR    RR      ....
BBBBBBBB    AA      AA  SSSSSSSS    PP          000000    WW      WW  RR      RR  RR      RR    ....
BBBBBBBB    AA      AA  SSSSSSSS    PP          000000    WW      WW  RR      RR  RR      RR    ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  BAS$POWRR                ; BASIC real ** real routine
0000     2          .IDENT  /1-005/          ; File: BASPOWRR.MAR Edit: RNH1005
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; FACILITY: Basic Support Library
0000    31 ;
0000    32 ; ABSTRACT:
0000    33 ;
0000    34 ;       This module contains entry points to support exponentiation
0000    35 ;       (** or ^) in BASIC-PLUS-2 for FLOATING ** FLOATING.
0000    36 ;
0000    37 ; ENVIRONMENT: User Mode, AST Reentrant
0000    38 ;
0000    39 ;--
0000    40 ; AUTHOR: R. Will       , CREATION DATE: 22-NOV-78
0000    41 ;
0000    42 ; MODIFIED BY:
0000    43 ;
0000    44 ; R. Will,       : VERSION 01
0000    45 ; 1-01  - Original
0000    46 ; 1-02  - Fix comments, change BRW to JMP.  RW 7-Dec-78
0000    47 ; 1-003 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000    48 ; 1-004 - Redo case analysis for base leg 0 for compatability
0000    49 ;          with the PDP-11.  JBS 24-APR-1979
0000    50 ; 1-005 - Change shared external references to G^ RNH 25-Sep-81
```

```
                        .SBTTL  DECLARATIONS
                ;
                ; INCLUDE FILES:
                ;

                ;
                ; EXTERNAL DECLARATIONS:
                ;
                        .DSABL  GBL                     ; Prevent undeclared
                                                        ; symbols from being
                                                        ; automatically global.

                        .EXTRN  OTS$POWRR               ; OTS$ real ** real exponentiation
                        .EXTRN  OTS$POWRJ               ; OTS$ real ** int exponentation
                        .EXTRN  BAS$K_DIVBY_ZER         ; Divide by Zero
                        .EXTRN  BAS$K_ILLARGLOG         ; Illegal argument in LOG
                        .EXTRN  BAS$$STOP               ; Error reporting routine

                ;
                ; MACROS:
                ;

                ;
                ; EQUATED SYMBOLS:
                ;

                ;
                ; OWN STORAGE:
                ;

                ;
                ; PSECT DECLARATIONS:
                ;
00000000        .PSECT _BAS$CODE PIC, USR, CON, REL, LCL, SHR, -
                        EXE, RD, NOWRT, LONG
```

```
                        0000     89          .SBTTL  BAS$POWRR  - BASIC floating ** floating
                        0000     90 ;++
                        0000     91 ; FUNCTIONAL DESCRIPTION:
                        0000     92 ;
                        0000     93 ;       This routine takes BASE ** EXP, using the following table
                        0000     94 ;       for unusual cases:
                        0000     95 ;
                        0000     96 ;       BASE > 0                            Call OTS$POWRR, normal case.
                        0000     97 ;       BASE = 0, EXP > 0                   Return 0.0.
                        0000     98 ;       BASE = 0, EXP = 0                   Return 1.0.
                        0000     99 ;       BASE = 0, EXP < 0                   Error: divide by zero
                        0000    100 ;       BASE < 0, EXP even integer          Call OTS$POWRJ with -BASE
                        0000    101 ;       BASE < 0, EXP odd integer           Call OTS$POWRJ with -BASE, negate result
                        0000    102 ;       BASE < 0, EXP not integer           Error: illegal argument in LOG.
                        0000    103 ;
                        0000    104 ; CALLING SEQUENCE:
                        0000    105 ;
                        0000    106 ;       CALL result.wf.v = BAS$POWRR (base.rf.v, exponent.rf.v)
                        0000    107 ;
                        0000    108 ; INPUT PARAMETERS:
                        0000    109 ;
00000004                0000    110         base = 4
00000008                0000    111         exponent = 8
                        0000    112 ;
                        0000    113 ; IMPLICIT INPUTS:
                        0000    114 ;
                        0000    115 ;       NONE
                        0000    116 ;
                        0000    117 ; OUTPUT PARAMETERS:
                        0000    118 ;
                        0000    119 ;       NONE
                        0000    120 ;
                        0000    121 ; IMPLICIT OUTPUTS:
                        0000    122 ;
                        0000    123 ;       NONE
                        0000    124 ;
                        0000    125 ; FUNCTION VALUE:
                        0000    126 ; COMPLETION CODES:
                        0000    127 ;
                        0000    128 ;       floating result of exponentiation
                        0000    129 ;
                        0000    130 ; SIDE EFFECTS:
                        0000    131 ;
                        0000    132 ;       Will signal Divide By Zero or Illegal argument in LOG if its
                        0000    133 ;       arguments are bad, and OTS$POWRR and OTS$POWRJ may also signal.
                        0000    134 ;
                        0000    135 ;--
                        0000    136
                  0000' 0000    137 BAS$POWRR::     .MASK OTS$POWRR             ; Entry point
                        0002    138                                             ; Since this routine uses no
                        0002    139                                             ; registers and usually transfers
                        0002    140                                             ; control to OTS$POWRR, we copy
                        0002    141                                             ; its register save mask and then
                        0002    142                                             ; JMP past its save mask and only
                        0002    143                                             ; save the registers once
04 AC     53            0002    144         TSTF    base(AP)                    ; Test base relationship to zero
   06     15            0005    145         BLEQ    1$                          ; If base leq 0, do case analysis
```

```
                00000002'GF   17  0007   146          JMP     G^OTS$POWRR+2           ; Transfer control to the OTS$
                              17  000D   147                                          ; routine to do exponentiation
                              17  000D   148 ;+
                              17  000D   149 ; Come here if the base is less than or equal to zero.  We must filter
                              17  000D   150 ; several special cases, as described above.
                              17  000D   151 ;-
                          2E  13  000D   152 1$:     BEQL    4$                      ; Branch if base = 0
    50   50   08   00   08 AC  54  000F   153          EMODF   exponent(AP), #0, #1, R0, R0
                          1A  12  0016   154          BNEQ    3$                      ; Branch if exponent is not integer
                              12  0018   155 ;+
                              12  0018   156 ; The base is less than zero and the exponent is an integer.
                              12  0018   157 ; BASIC defines this as working the same way as if an integer was
                              12  0018   158 ; in the expression (making a floating variable which happens to
                              12  0018   159 ; contain an integer value equivalent to an integer variable).
                              12  0018   160 ;-
                50   08 AC   4A  0018   161          CVTFL   exponent(AP), R0        ; Convert exponent to integer
                        50    DD  001C   162          PUSHL   R0                      ; Save for even/odd test
                        50    DD  001E   163          PUSHL   R0                      ; Stack as parameter to OTS$POWRJ
                     7E   04 AC   52  0020   164          MNEGF   base(AP), -(SP)         ; Stack -base also
           00000000'GF    02    FB  0024   165          CALLS   #2, G^OTS$POWRJ         ; Call integer power routines
                     03 8E        E9  002B   166          BLBC    (SP)+,2$                ; Branch if exponent even
                50   50   52  002E   167          MNEGF   R0, R0                  ; Exponent odd, negate the result
                        04    0031   168 2$:     RET                             ; and return with it.
                              0032   169 ;+
                              0032   170 ; Come here if the base is less than zero but the exponent is not
                              0032   171 ; an integer.  BASIC defines this as an error.
                              0032   172 ;-
                     7E   00'8F   9A  0032   173 3$:     MOVZBL  #BAS$K_ILLARGLOG, -(SP) ; Illegal Argument in LOG
           00000000'GF    01    FB  0036   174          CALLS   #1, G^BAS$$STOP         ; Never return.
                              003D   175 ;+
                              003D   176 ; Come here if the base is equal to zero.  The value we return depends
                              003D   177 ; upon the sign of the exponent.
                              003D   178 ;-
                08 AC    53  003D   179 4$:     TSTF    exponent(AP)            ; Test the exponent against zero
                   09    19  0040   180          BLSS    6$                      ; Branch if exponent lss 0
                   03    13  0042   181          BEQL    5$                      ; Branch if exponent is 0
                              0044   182 ;+
                              0044   183 ; Come here if the base is zero and the exponent is greater than zero.
                              0044   184 ; BASIC defines this as 0.0.
                              0044   185 ;-
                50    D4    0044   186          CLRF    R0                      ; R0 = 0.0
                        04    0046   187          RET                             ; Return to caller
                              0047   188 ;+
                              0047   189 ; Come here if the base is zero and the exponent is zero.  BASIC defines
                              0047   190 ; this as 1.0.
                              0047   191 ;-
             50   08   50    0047   192 5$:     MOVF    #1, R0                  ; R0 = 1.0
                        04    004A   193          RET                             ; Return to caller.
                              004B   194 ;+
                              004B   195 ; Come here if the base is zero and the exponent is less than zero.
                              004B   196 ; BASIC defines this as an error.
                              004B   197 ;-
          7E   00'8F   9A  004B   198 6$:     MOVZBL  #BAS$K_DIVBY_ZER, -(SP) ; Divide by zero
00000000'GF    01    FB  004F   199          CALLS   #1, G^BAS$$STOP         ; Report error, never return.
                              0056   200 ;
                              0056   201          .END
```

```
BAS$$STOP           ********    X   00
BAS$K_DIVBY_ZER     ********    X   00
BAS$K_ILLARGLOG     ********    X   00
BAS$POWRR           00000000 RG     01
BASE            =   00000004
EXPONENT        =   00000008
OTS$POWRJ           ********    X   00
OTS$POWRR           ********    X   00
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_BAS$CODE	00000056 (86.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	38	00:00:00.08	00:00:00.58
Command processing	127	00:00:00.49	00:00:02.20
Pass 1	69	00:00:00.48	00:00:01.26
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	48	00:00:00.40	00:00:00.82
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	289	00:00:01.50	00:00:04.90

The working set limit was 750 pages.
2270 bytes (5 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 8 non-local and 6 local symbols.
201 source lines were read in Pass 1, producing 8 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

```
MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:BASPOWRR/OBJ=OBJ$:BASPOWRR MSRC$:BASPOWRR/UPDATE=(ENH$:BASPOWRR)
```

BASOPEN
LIS

BASPOWJJ
LIS

BASPOS
LIS

BASPOWDJ
LIS

BASOPENDE
LIS

BASPOWGG
LIS

BASPOWHH
LIS

BASPOWRJ
LIS

BASPOWTI
LIS

BASPURIOB
LIS

BASPOWDO
LIS

BASOPENZE
LIS

BASPOWOR
LIS

BASPOWG
LIS

BASPOWRD
LIS

BASPOWH
LIS

BASPOWRR
LIS